



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/773,068	02/04/2004	Steve Elmer	AOL0134	6042
22862 7590 12/22/2008				
GLENN PATENT GROUP				
3475 EDISON WAY, SUITE L				
MENLO PARK, CA 94025				
EXAMINER				
ADAMS, CHARLES D				
ART UNIT		PAPER NUMBER		
2164				
MAIL DATE		DELIVERY MODE		
12/22/2008		PAPER		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

### Office Action Summary

**Application No.**

10/773,068

**Applicant(s)**

ELMER, STEVE

**Examiner**

CHARLES D. ADAMS

**Art Unit**

2164

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 28 August 2008.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-3, 7-13, 17-23 and 27-30 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-3, 7-13, 17-23 and 27-30 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/06)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

## DETAILED ACTION

### *Remarks*

1. In response to communications filed on 28 August 2008, claims 1, 11, and 21 are amended. Claims 1-3, 7-13, 17-23, and 27-30 are pending in the application.

### *Specification*

2. Claim 11 is objected to because of the following informalities:

The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o). Correction of the following is required: The specification does not contain a processor.

. Appropriate correction is required.

### *Claim Rejections - 35 USC § 103*

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-3, 11-13, and 21-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glerum et al. (US Patent 6,652,267) in view of Marullo et al. (US Patent 6,185,701), and further in view of Dutta et al. (US Patent 6,918,066), and further in view of Miller et al. (US Pre-Grant Publication 2007/0234217).

As to claim 1, Glerum et al. teaches a method of testing browser software in a computer environment (see Abstract and 1:38-57), the method comprising the steps of:

Glerum et al. does not teach generating a list of URLs (Universal Resource Location) using a web crawler;

Marullo et al. teaches generating a list of URLs (Universal Resource Location) using a web crawler (see 13:62-14:14).

Glerum et al. as modified teaches applying a browser test (see Marullo et al. 8:22-45), wherein said browser test script automatically instructs a first browser program containing said browser software to load and render web pages according to the list of URLs, wherein said browser test script tests said browser software over a plurality of applications at sites contained within the list of URLs (see Marullo et al. 8:22-45);

Detecting one or more browser rendering errors of said first browser program using the web pages (see Glerum et al. 1:38-57 and 4:63-5:3)

Glerum et al. as modified does not teach by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program, wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program

Dutta et al. teaches by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program (see 7:23-35 and 7:50-65), wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the

browser program (see 7:50-65. The scorecard evaluates how well the various browsers displayed a page based on an internal representation of how the page was displayed, “this scorecard compares the web site as it is displayed on each browser to a standard set of criteria that includes accessibility of the web site, the percentage of the content of the web site displayed by each browser, the load time of the web site on each browser, and the compatibility of scripts of the web site across different browsers”)

Glerum et al. as modified does not explicitly teach wherein the internal representation includes a list of objects to be displayed for the web page and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to ensure that there is no glitch in interpreting the information defining the web page;

Miller et al. teaches wherein the internal representation includes a list of objects to be displayed for the web page and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to ensure that there is no glitch in interpreting the information defining the web page (see paragraphs [0024], [0082]-[0083], and [0088]-[0094]. The user can select images, text, and tables. Objects will be stored identifying these items along with their attributes in a script. Upon playback, if the information in the script does not match the internal representation of the webpage an error results(see paragraph [0024]));

Glerum et al. as modified teaches:

and wherein one or more errors are detected when the representation of rendering results of the first browser program does not match the representation of

rendering results of the second browser program (see Dutta et al. 8:9-11, 8:41-55, and 8:65-9:14 and Miller et al. paragraphs [0024], [0089], [0091], and [0093]); and

Automatically storing information about said one or more errors (see Glerum et al. 4:63-5:3);

Wherein said step of applying a browser test script is performed while said first browser program is under development and prior to distribution (see Glerum et al. 4:63-5:3 and 8:61-9:3).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Marullo et al., since Marullo et al. teaches that "by providing for the aforementioned automated client-based web universal resource (link) extraction tool, such automation avoids the inadequacies associated with user testing and intervention wherein manual users might otherwise be required to request pages, view document source, and document all of the links (assuming they were found without error) associated with the HTML pages. The getlinks subsystem accordingly automatically finds all links on each page, and moreover formats the output data for use by the other test tools" (see 14:32-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have further modified Glerum et al. by the teaching of Dutta et al., since Dutta et al. teaches that "In addition to testing the web site on the various browsers to determine the effectiveness of each browser, it is also desirable to have a technique that can compare each browser's execution of the web site to a set of criteria established by the web designed" (see 3:15-20).

It would also have been obvious to have further modified Glerum et al. by the teachings of Miller et al., as Miller et al. is also related to testing browser software to determine its usability and whether or not it can correctly render webpages. In addition to this, Miller et al. teaches "broadly speaking, the invention relates to improved approaches for testing websites. According to one aspect of the invention website testing is performed in a browser environment. As such, information for testing can be obtained from a browser that is able to perform automated analysis and testing of websites" (see paragraph [0010]).

As to claim 11, Glerum et al. teaches an apparatus of testing browser software in a computer environment (see Abstract and 1:38-57), comprising:

at least one processor programmed to provide (see Abstract and 3:37-58):

A plurality of software modules (see Abstract and 1:38-57), said plurality of software modules including:

Glerum et al. does not teach a module for generating a list of URLs (Universal Resource Location) using a web crawler;

Marullo et al. teaches a module for generating a list of URLs (Universal Resource Location) using a web crawler (see 13:62-14:14);

Glerum et al. as modified teaches a browser test script module (see Marullo et al. 8:22-45), wherein said browser test script module automatically instructs a first browser program containing said browser software to load and render web pages according to the list of URLs, wherein said browser test script tests said browser software over a

plurality of applications at sites contained within the list of URLs (see Marullo et al. 8:22-45);

A module for detecting one or more browser rendering errors of said first browser program using the web pages (see Glerum et al. 1:38-57 and 4:63-5:3)

Glerum et al. as modified does not teach by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program, wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program

Dutta et al. teaches by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program (see 7:23-35 and 7:50-65) wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program (see 8:9-11, 8:41-55, and 8:65-9:14)

Glerum et al. as modified does not teach wherein the internal representation includes a list of objects to be displayed for the web pages and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to ensure that there is no glitch in interpreting the information defining the webpage

Miller et al. teaches wherein the internal representation includes a list of objects to be displayed for the web pages and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to



ensure that there is no glitch in interpreting the information defining the webpage (see paragraphs [0024], [0082]-[0083], and [0088]-[0094]); and

Glerum et al. as modified teaches:

wherein one or more errors are detected when the representation of rendering results of the first browser program does not match the representation of rendering results of the second browser program (see Dutta et al. 8:9-11, 8:41-55, and 8:65-9:14 and Miller et al. paragraph [0024])

A module for automatically storing information about said one or more errors (see Glerum et al. 4:63-5:3);

Said apparatus further comprising at least one computer, wherein said plurality of software modules are executed on said at least one computer (see Glerum et al. Abstract, 1:38-57, and 3:36-58)

Wherein said first browser program is under development prior to distribution of said browser program (see Glerum et al. 4:63-5:3 and 8:61-9:3).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Marullo et al., since Marullo et al. teaches that "by providing for the aforementioned automated client-based web universal resource (link) extraction tool, such automation avoids the inadequacies associated with user testing and intervention wherein manual users might otherwise be required to request pages, view document source, and document all of the links (assuming they were found without error) associated with the HTML pages. The

getlinks subsystem accordingly automatically finds all links on each page, and moreover formats the output data for use by the other test tools" (see 14:32-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have further modified Glerum et al. by the teaching of Dutta et al., since Dutta et al. teaches that "In addition to testing the web site on the various browsers to determine the effectiveness of each browser, it is also desirable to have a technique that can compare each browser's execution of the web site to a set of criteria established by the web designed" (see 3:15-20).

It would also have been obvious to have further modified Glerum et al. by the teachings of Miller et al., as Miller et al. is also related to testing browser software to determine its usability and whether or not it can correctly render webpages. In addition to this, Miller et al. teaches "broadly speaking, the invention relates to improved approaches for testing websites. According to one aspect of the invention website testing is performed in a browser environment. As such, information for testing can be obtained from a browser that is able to perform automated analysis and testing of websites" (see paragraph [0010]).

As to claim 21, Glerum et al. teaches a program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform a method for testing a browser software in a computer environment (see Abstract and 1:38-57), the method comprising the steps of:

Glerum et al. does not teach generating a list of URLs (Universal Resource Location) using a web crawler;

Marullo et al. teaches generating a list of URLs (Universal Resource Location) using a web crawler (see 13:62-14:14);

Glerum et al. as modified teaches applying a browser test script (see Marullo et al. 8:22-45), wherein said browser test script automatically instructs a first browser program containing said browser software to load and render web pages according to the list of URLs, wherein said browser test script tests said browser software over a plurality of applications at sites contained within the list of URLs (see Marullo et al. 8:22-45);

Detecting one or more browser rendering errors of said first browser program using the web pages (see Glerum et al. 1:38-57 and 4:63-5:3)

Glerum et al. as modified does not teach by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program, wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program;

Dutta et al. teaches by comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program (see 7:23-35 and 7:50-65), wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program (see Dutta et al. 8:9-11, 8:41-55, and 8:65-9:14)

Glerum et al. as modified does not teach wherein the internal representation includes a list of objects to be displayed for the web page and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to ensure that there is no glitch in interpreting the information defining the web page

Miller et al. teaches wherein the internal representation includes a list of objects to be displayed for the web page and the attributes of the objects, and wherein the attributes and the objects that are uniquely defined by the web page are compared to ensure that there is no glitch in interpreting the information defining the web page (see paragraphs [0024], [0082]-[0083], and [0088]-[0094]))

Glerum et al. as modified teaches:  
and wherein one or more errors are detected when the representation of rendering results of the first browser program does not match the representation of rendering results of the second browser program (see Dutta et al. 8:9-11, 8:41-55, and 8:65-9:14 and Miller et al. paragraph [0024])

Automatically storing information about said one or more errors (see Glerum et al. 4:63-5:3);

Wherein said step of applying a browser test script is performed while said first browser program is under development and prior to distribution (see Glerum et al. 4:63-5:3 and 8:61-9:3).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Marullo

et al., since Marullo et al. teaches that "by providing for the aforementioned automated client-based web universal resource (link) extraction tool, such automation avoids the inadequacies associated with user testing and intervention wherein manual users might otherwise be required to request pages, view document source, and document all of the links (assuming they were found without error) associated with the HTML pages. The getlinks subsystem accordingly automatically finds all links on each page, and moreover formats the output data for use by the other test tools" (see 14:32-43).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have further modified Glerum et al. by the teaching of Dutta et al., since Dutta et al. teaches that "In addition to testing the web site on the various browsers to determine the effectiveness of each browser, it is also desirable to have a technique that can compare each browser's execution of the web site to a set of criteria established by the web designed" (see 3:15-20).

It would also have been obvious to have further modified Glerum et al. by the teachings of Miller et al., as Miller et al. is also related to testing browser software to determine its usability and whether or not it can correctly render webpages. In addition to this, Miller et al. teaches "broadly speaking, the invention relates to improved approaches for testing websites. According to one aspect of the invention website testing is performed in a browser environment. As such, information for testing can be obtained from a browser that is able to perform automated analysis and testing of websites" (see paragraph [0010]).

As to claims 2, 12, and 22, Glerum et al. as modified teaches wherein the one or more errors include a crash of the first browser program in rendering one of the web pages (see 1:38-57).

As to claims, 3, 13, and 23, Glerum et al. as modified teaches the claim upon which this claim is dependent.

Glerum et al. as modified teaches said browser test script automatically instructs said second browser program load and render the web pages (see Dutta et al. 7:23-35 and Marullo et al. 8:22-45);

5. Claims 7, 17, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glerum et al. (US Patent 6,6529,267) in view of Marullo et al. (US Patent 6,185,701), in view of Dutta et al. (US Patent 6,918,066), in view of Miller et al. (US Pre-Grant Publication 2007/0234217), and further in view of Castro ("HTML FOR THE WORLD WIDE WEB").

Glerum et al. as modified teaches the claims upon which these claims are dependent.

Glerum et al. does not explicitly teach wherein the internal representation of the web page comprises attributes of the web page, including:

- A background color;
- A number of columns of a table; and
- A number of rows of a table.

Castro teaches wherein the internal representation of the web page comprises attributes of the web page, including:

A background color (see page 228);

A number of columns of a table (see pages 228 and 233. A number of columns are shown in the HTML code); and

A number of rows of a table (see pages 228 and 233. A number of rows are shown in the HTML code).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. to have included the teaching of Castro to interpret common HTML code when rendering the web page, since interpreting a web page and rendering the web page based on the tags found within it was well known to anyone of ordinary skill in the art at the time the invention was made. It would have been an obvious test as to how well a browser worked by determining whether or not the browser could render 'table' and 'background color' HTML tags.

6. Claims 8, 18, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glerum et al. (US Patent 6,6529,267) in view of Marullo et al. (US Patent 6,185,701), in view of Dutta et al. (US Patent 6,918,066), in view of Miller et al. (US Pre-Grant Publication 2007/0234217), and further in view of Shindo (US Patent 6,865,592).

Glerum et al. as modified teaches the claim upon which these claims depend.

Glerum et al. as modified does not teach automatically restarting the first browser program after a crash of the first browser program in rendering one of the web pages.

Shindo teaches automatically restarting the first browser program after a crash of the first browser program in rendering one of the web pages (see Shindo 11:15-23).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Shindo, since Shindo teaches that "if a failure occurs due to the Web environment on the automatic transaction apparatus side, such as halting of the Web browser, or if a failure occurs due to the Web environment on the Web server side, such as shut-down of the server or congestion on the network, the automatic transaction apparatus cannot download applications required to operate. Therefore the automatic transaction apparatus halts the process. If the automatic transaction apparatus stops, customers cannot be serviced" (see 1:31-39).

7. Claim 9-10, 19-20, and 29-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Glerum et al. (US Patent 6,6529,267) in view of Marullo et al. (US Patent 6,185,701), in view of Dutta et al. (US Patent 6,918,066), in view of Miller et al. (US Pre-Grant Publication 2007/0234217), and further in view of Garcia-Chiesa (US Pre-Grant Publication 2002/0099723).

As to claims 9, 19, and 29, Glerum et al. as modified teaches the claim upon which these claims depend.



Glerum et al. as modified teaches does not teach further comprising the step of avoiding duplicated visits to a same URL.

Garcia-Chiesa teaches further comprising the step of avoiding duplicated visits to a same URL (see Garcia-Chiesa paragraph [0010]. "Further, the methods the present invention generate lists of unique URLs that are marked each of them as static, thus the engines do not need to follow ANY non-static link. Plus, the list that follows is deduplicated, optimized and sanitized").

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Garcia-Chiesa, since Garcia-Chiesa teaches that "Further more the techniques include the elimination of possible crawling loops that due to minor differences in the emitted URLs format could otherwise be undetected by crawlers not specifically aware of the non-materiality of these subtle syntactic differences" (see paragraph [0043]).

As to claim 10, 20, and 30, Glerum et al. as modified teaches the claims upon which these claims are dependent.

Glerum et al. does not teach wherein a number of URLs are removed from the URLs reported by the web crawler to generate the list of URLs (see Garcia-Chiesa paragraph [0010]. Removing duplicates will remove URLs from the list).

Garcia-Chiesa teaches wherein a number of URLs are removed from the URLs reported by the web crawler to generate the list of URLs (see Garcia-Chiesa paragraph [0010]. Removing duplicates will remove URLs from the list).

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified Glerum et al. by the teaching of Garcia-Chiesa, since Garcia-Chiesa teaches that "Further more the techniques include the elimination of possible crawling loops that due to minor differences in the emitted URLs format could otherwise be undetected by crawlers not specifically aware of the non-materiality of these subtle syntactic differences" (see paragraph [0043]).

### ***Response to Arguments***

8. Applicant's arguments filed 28 August 2008 have been fully considered but they are not persuasive.

In regards to the independent claims, Applicant argues that "There is no teaching or suggestion anywhere in Marullo of applying a browser test script that automatically instructs a browser program to load and render web pages". In response to this argument, it is noted that the cited teachings of Marullo et al. teach instructing a virtual browser to automatically test and verify web server applications. It is noted that column 10, lines 1-50 show an example script that clearly contains addresses to websites and arguments to test on those websites.

Applicant continues this argument by stating "Accordingly, what is described by Marullo, is 'a virtual browser application' that automatically tests web servers and/or web applications. The Office may say that a "virtual browser" is a browser. However,

this would be incorrect because, as Marullo teaches, the webrunner subsystem is a testing tool that replaces a browser in the web server application testing process, because, as Marullo also teaches at 2:12-52 and 3:29-33 and 6:55-59, the use of conventional browsers, even automatically driven, is unsuitable for the testing of web servers and web server applications. Marullo teaches that the use of a browser for testing web server application is unsuitable because it produces inconsistent results that are determined by the browser. In addition, Marullo teaches that the testing of a browser application and testing of a web server application are distinct tasks, and that what is needed for web server testing is a software tool specifically tailored for web server application testing – not a conventional browser. Thus, the webrunner subsystem is not a browser itself, nor is it a test script that instructs a browser.”

In response to this argument, Examiner notes that a “virtual browser” is still a type of browser. The word “virtual” discloses the type of browser. Examiner sees no qualifying limitations in the claims that would prohibit the use of “virtual browsers”. The claims are limited to a “browser program.” It is noted that, since the “virtual browser” is a browser program, the Webrunner system does contain a browser itself, and a test script that instructs a browser. It is also noted that the claim was rejected at this point by Glerum et al. in view of Marullo et al., and Glerum et al. clearly teaches testing a browser and reporting failure errors. Marullo et al.’s objections in 2:12-21 are directed not to failures of the browser, but failures of the testing process occurring in recording GUI and live testers missing all possible inputs/outputs. The same is true in the drawbacks Marullo et al. mentioned in 3:29-33. In 6:55-59, Marullo et al. objects to prior

art solutions employing conventional browsers, due to compatibility and performance issues, and recording user input. It is noted that these are not drawbacks of the browser itself, but the manner in which it was used and the prior art system which employed the conventional browsers. It is also noted that the claims are not limited to a "conventional browser," but merely a "browser program." A "browser program" is any program that functions as a browser, which Marullo et al. clearly teaches.

In addition to this, Dutta et al., which is relied upon later, teaches testing using emulations of "conventional browsers." and Glerum et al. teaches testing "conventional browsers".

Applicants also state "Marullo additionally teaches away from the use of live testers to manually test web server applications, at least at 2:3-21, citing a number of problems: the slowness of the process, the difficulty of employing large numbers of testers, and the difficulty of testing a web server application thoroughly and consistently." In response to this argument, it is noted that Marullo et al.'s proposed solution to these problems is through automation. It is also noted that the claims are not limited to requiring live testers.

Applicants argue that "the remaining references add nothing to Marullo. Dutta describes a method of testing a web server application on different browsers. however, the process, as described by Dutta, is at least partially manual, wherein the user must select a file or a web page and supply it to the browsers being used to perform the test.

Additionally, there is no teaching or suggestion in Dutta of using a browser test script to test browser software over a plurality of applications at sites contained within the list of URL". It is noted that Marullo et al. teaches using a browser test script to test browser software over a plurality of applications at sites contained within a list of URLs (see 8:22-45 and 10: 1-50). Applicant continues "As previously mentioned, Dutta has nothing to do with testing browser software. Dutta is directed to testing web server applications. Additionally, it appears that Dutta can only test one web application at a time. Dutta does not appear to be able to test browser software over a plurality of applications at sites contained within a list of URLs." As noted above, Marullo et al. is relied upon for this teaching.

Applicant also argues that Miller et al. does not teach applying a browser test script. In response to this, it is noted that Marullo et al. is relied upon for this teaching, not Miller et al.

Applicant argues that "there is no teaching or suggestion whatsoever in Glerum of detecting a browser's rendering errors. Glerum describes a method and system for reporting program failures. Glerum defines program failures as either crashes or set-up failures. In addition, Glerum describes that the system also reports a 'problem encountered during in-house testing of a program module.' However, the only example Glerum provides of such a testing' problem is at 8:61-63." In response to this argument, it is noted that a "crash" is a "rendering error". Claims 2, 12, and 22 further limit the one

or more errors to include "a crash of the first browser program in rendering one of the web pages".

Applicant argues that "the remaining references add nothing to compensate for Glerum's deficiencies". It is noted that none of the remaining references are relied upon to teach "detecting one or more errors in rendering of said first browser program using the web pages".

Applicant argues that "the scorecard parameters described [in Dutta] represent performance criteria for the web page". In response to this argument, it is noted that the limitation reads "comparing a representation of rendering results of the first browser program to a representation of rendering results of a second browser program, wherein a representation of rendering results of a browser program comprises an internal representation of a web page as interpreted by the browser program". It is noted that the scorecard of Dutta et al. evaluates how well the various browsers displayed a page, based on an internal representation of the page. It is noted that the scorecard "compares the website as it is displayed on each browser to a standard set of criteria that includes accessibility of the website, the percentage of the content of the web site displayed by each browser, the load time of the website on each browser, and the compatibility of scripts of the web site across different browsers" (see Dutta et al. 7:52-57).

Applicant continues "thus, as described in Dutta, what is being evaluated is the website, and not the browser. As taught in Marullo, web site evaluation and web

browser evaluation are distinct tasks requiring different tools. They are not indistinguishable or interchangeable, as the Office has previously maintained. Dutta's system has nothing to do with tracking browser rendering errors. Dutta's scorecard, as amply demonstrated above, has nothing to do with tracking browser errors. It is configured for assessing deficiencies of the website - which, as Marullo teaches, is an entirely different task from evaluating the web browser."

In response to this argument, it is noted that the scorecard output of Dutta et al. evaluates browsers. It is an output of how well multiple browsers can display a website. A user can then edit the website so that a browser displays the website more accurately or completely. It is also noted that the system of Dutta et al. can track browser rendering errors, in that it can report whether or not a browser accurately displayed a script, and how much of the content was displayed on each browser.

Applicant adds "additionally, the ordinarily skilled practitioner would have no motivation to combine the teachings of Marullo/Glerum and Dutta because Dutta's method is largely under human control and is performed on an individual web server application. As above, Marullo explicitly criticizes human-driven testing as being overly time-consuming, expensive, inconsistent, and lacking in thoroughness. Accordingly, the ordinarily-skilled practitioner would not be lead to Dutta to remedy the deficiencies of Marullo/Glerum." In response to this argument, it is noted that the rendering of web pages in Dutta et al. is not under human control. The webpages are rendered in separate browser emulators, and a scorecard is generated based on how the webpages were rendered. These tasks are run by a computer. It is noted that in Marullo et al., the

script automates a computer to test webpages. In addition to this, the tested webpages are run individually, in a sequence. Thus, a skilled practitioner would have had motivation to combined the teachings of Marullo et al., Glerum et al., and Dutta et al.

Applicant argues that "while the cited paragraphs of Miller appear to resemble the claimed subject mater, the fact remains that Miller is directed to cataloging the deficiencies of a web application, which as Marullo has established, is a separate and distinct task from evaluating a browser. There is no teaching or suggesting anywhere in Miller that any step of Miller's methodology is directed to detecting browser rendering errors." In response to this argument, Examiner notes that paragraphs [0082]-[0083] and [0088]-[0094] in Miller et al. explicitly deal with detecting rendering errors in browsers during visits to a website.

Applicant argues that "additionally, the combination is improper because the Office cites no motivation for combining the teachings of Marullo and Glerum. The Office is respectfully reminded that, even in the post-KSR climate, establishing prima facie obviousness requires that the Office identify a motivation for combining the reference teachings." In response to this argument, it is noted that motivation for combining Glerum et al. and Marullo et al. was provided on page 6 of the Office Action mailed 29 May 2008.



In regards to claims 7-10, 17-20, and 27-30, Applicant argues that the Examiner has combined an excessive number of references. In response to the argument that the examiner has combined an excessive number of references, reliance on a large number of references in a rejection does not, without more, weigh against the obviousness of the claimed invention. See *In re Gorman*, 933 F.2d 982, 18 USPQ2d 1885 (Fed. Cir. 1991).

Applicant argues that "Additionally, Castro has nothing to do with an internal representation of a web page. Rather Castro is describing web page attributes as they are found in the source code for the page." In response to this argument, it is noted that the source code of the page is an internal representation of a page. It is the content of the page, and the evaluation system of Dutta et al. teaches to evaluate what percentage of the content of the page was displayed.

### ***Conclusion***

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHARLES D. ADAMS whose telephone number is (571)272-3938. The examiner can normally be reached on 8:30 AM - 5:00 PM, M - F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/C. D. A./  
Examiner, Art Unit 2164

/Charles Rones/  
Supervisory Patent Examiner, Art Unit 2164